



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/796,613	03/08/2004	Leela S. Tamma	MSI-1880US	1791
22801	7590	11/28/2007	EXAMINER	
LEE & HAYES PLLC			CHEN, QING	
421 W RIVERSIDE AVENUE SUITE 500			ART UNIT	PAPER NUMBER
SPOKANE, WA 99201			2191	
		MAIL DATE	DELIVERY MODE	
		11/28/2007	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	10/796,613	TAMMA ET AL.
Examiner	Art Unit	
Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

1) Responsive to communication(s) filed on 07 September 2007.  
 2a) This action is **FINAL**.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

4) Claim(s) 1-31 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-31 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

1) Notice of References Cited (PTO-892)  
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3) Information Disclosure Statement(s) (PTO/SB/08)  
 Paper No(s)/Mail Date \_\_\_\_\_

4) Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_

5) Notice of Informal Patent Application  
 6) Other: \_\_\_\_\_

### **DETAILED ACTION**

1. This Office action is in response to the amendment filed on September 7, 2007.
2. **Claims 1-31** are pending.
3. **Claims 3-7, 10, 12, 20, 23, and 28-31** have been amended.
4. The objection to the drawings is withdrawn in view of Applicant's amendments to the specification.
5. The objections to the specification are withdrawn in view of Applicant's amendments to the specification.
6. The objections to Claims 7, 10, 23, and 31 are withdrawn in view of Applicant's amendments to the claims.
7. The 35 U.S.C. § 101 rejections of Claims 12, 20-22, and 29-31 are withdrawn in view of Applicant's amendments to the claims.

#### *Response to Amendment*

#### *Claim Rejections - 35 USC § 112*

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
9. **Claims 3 and 4** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

**Claims 3 and 4** recite the limitation “the first version.” There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “a first version” for the purpose of further examination.

***Claim Rejections - 35 USC § 103***

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. **Claims 1-6, 8, 10-20, and 23-31** are rejected under 35 U.S.C. 103(a) as being unpatentable over **US 6,970,876 (hereinafter Hotti)** in view of **US 6,415,299 (hereinafter Baisley)**.

As per **Claim 1**, Hotti discloses:

- automatically determining a first set of data definition language (DDL) scripts associated with the particular version of the relational database (*see Column 2: 9-10, “Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.”; Column 4: 49-58, “Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation*

*scripts. "; Column 7: 51-54, "The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server. "); and*

- automatically determining a second set of data manipulation language scripts associated with the particular version of the relational database (see Column 2: 9-10, *"Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node. "; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Définition Language) scripts, or any other data manipulation scripts. "; Column 7: 51-54, "The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server. ").*

However, Hotti does not disclose:

- generating an installation file comprising a union of the first set and the second set.

Baisley discloses:

- generating an installation file comprising a union of the first set and the second set (see Column 2: 14-19, *"The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include generating an installation file comprising a union of the first set and the second set. The

modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Hotti further discloses:

- wherein the particular version is associated with a first version in a sequence of one or more versions of the relational database (*see Column 2: 7-8, “Schema revision” is a snapshot version of a schema that is identifiable by logical name or version number. ”*).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Hotti further discloses:

- wherein the automatically determining a first set comprises extracting a filename from metadata associated with a first version, the filename associated with a file comprising a data definition language script (*see Column 6: 18-21, “The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system. ” and 53-55, “In step 302 the database schema is defined using configuration management application and stored to the schema management master database, step 303. ” and 63-66, “As part of the registration, the identification data, e.g. schema name, of the new application database node is sent to the configuration management master database node. ”*).

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Hotti further discloses:

- wherein the automatically determining a second set comprises extracting a filename from metadata associated with a first version, the filename associated with a file comprising a data manipulation language script (*see Column 6: 18-21, "The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system."* and 53-55, *"In step 302 the database schema is defined using configuration management application and stored to the schema management master database, step 303."* and 63-66, *"As part of the registration, the identification data, e.g. schema name, of the new application database node is sent to the configuration management master database node."*).

As per **Claim 5**, the rejection of **Claim 1** is incorporated; however, Hotti does not disclose:

- wherein the generating an installation file comprises copying a data definition language script from a script file associated with the first set into the installation file.

Baisley discloses:

- wherein the generating an installation file comprises copying a data definition language script from a script file associated with the first set into the installation file (*see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include wherein the generating an installation file comprises copying a data definition language script from a script file associated with the first set into the installation file. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 6**, the rejection of **Claim 1** is incorporated; however, Hotti does not disclose:

- wherein the generating an installation file comprises copying a data manipulation language script from a script file associated with the second set into the installation file.

Baisley discloses:

- wherein the generating an installation file comprises copying a data manipulation language script from a script file associated with the second set into the installation file (*see Column 2: 14-19, “The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include wherein the generating an installation file comprises copying a data manipulation language script from a script file associated with the second set into the installation file. The modification would

be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see *Baisley* – Column 1: 21-24).

As per **Claim 8**, the rejection of **Claim 1** is incorporated; and Hotti further discloses:

- wherein metadata exists that describes a sequence of multiple versions of the relational database where each version is an upgrade from a previous version, and the particular version is not a first version in the sequence (see *Column 2: 14 and 15*. “*Schema script publication*” is a system publication that contains the schema scripts of the database hierarchy.”; *Column 6: 18-21*, “*The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system.*”).

As per **Claim 10**, the rejection of **Claim 8** is incorporated; however, Hotti does not disclose:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version;

- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ; and

- determining the first set as the union of sets  $A_1, A_2, \dots, A_j$ .

Baisley discloses:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version (*see Figure 3; Column 5: 17-36, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model."*);

- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database; the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$  (*see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. "*); and

- determining the first set as the union of sets  $A_1, A_2, \dots, A_j$  (*see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version;
- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ; and
- determining the first set as the union of sets  $A_1, A_2, \dots, A_j$ .

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 11**, the rejection of **Claim 8** is incorporated; however, Hotti does not disclose:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;
- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object

when upgrading from version  $i - 1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ;

- iteratively extracting a set  $B_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version  $i - 1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ; and

- determining the second set  $C_j$  by determining:

$$C_2 = [A_1 \bigcup A_2] - B_2,$$

$$C_3 = [C_2 \bigcup A_3] - B_3,$$

$$C_4 = [C_3 \bigcup A_4] - B_4,$$

...

$$C_j = [C_{j-1} \bigcup A_j] - B_j.$$

Baisley discloses:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version (see *Figure 3; Column 5: 17-36*, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model.");

- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version  $i - 1$  of the relational database to version  $i$  of the relational

database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$  (see *Figure 3; Column 5: 17-36*, “*In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. ”);*

- iteratively extracting a set  $B_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a script to be executed when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$  (see *Figure 3; Column 5: 17-36*, “*In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. ”); and*

- determining the second set  $C_j$  by determining:

$$C_2 = [A_1 \bigcup A_2] - B_2,$$

$$C_3 = [C_2 \bigcup A_3] - B_3,$$

$$C_4 = [C_3 \bigcup A_4] - B_4,$$

...

$$C_j = [C_{j-1} \bigcup A_j] - B_j$$

(see *Column 2: 14-19*, “*The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ”).*

Official Notice is taken that it is old and well-known within the computing art to include a drop script as part of the DML. A query language often provides a “drop” command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- extracting a set  $A_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;
- iteratively extracting a set  $A_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ;
- iteratively extracting a set  $B_i$  comprising zero or more filenames from metadata associated with an  $i^{th}$  version of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version  $i-1$  of the relational database to version  $i$  of the relational database, where  $i$  varies incrementally from 2 to  $j$ , where the particular version is  $j$ ; and
- determining the second set  $C_j$  by determining:

$$\begin{aligned}C_2 &= [A_1 \bigcup A_2] - B_2, \\C_3 &= [C_2 \bigcup A_3] - B_3, \\C_4 &= [C_3 \bigcup A_4] - B_4, \\&\dots \\C_j &= [C_{j-1} \bigcup A_j] - B_j.\end{aligned}$$

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see *Baisley – Column 1: 21-24*).

As per **Claim 12**, the rejection of **Claim 1** is incorporated; and Hotti further discloses:

- One or more computer-readable media having computer-readable instructions recorded thereon which, when executed by a computer, cause the computer to implement the method as recited in claim 1 (see *Column 5: 1-5*, “*The invention also relates to a storage media comprising a stored, readable computer program, which is characterized in that the program comprises instructions for controlling a data management system or components thereof to implement the method according to the invention.*”).

As per **Claim 13**, Hotti discloses:

- determining a set  $A$  of data definition language (DDL) scripts that, when executed, resulting in DDL objects associated with version  $j$  of the relational database (see *Column 2: 9-10*, “*“Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.*”; *Column 4: 49-58*, “*Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data*

*manipulation scripts.”; Column 7: 51-54, “The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server.”;*

- determining a set  $B$  of data manipulation language (DML) scripts that, when executed, create DML objects that are associated with version  $j$  of the relational database, but that are not associated with version  $i$  of the relational database (see *Column 2: 9-10, “Schema script is a script that creates a schema or creates a new revision of an existing schema of a database node.”; Column 4: 49-58, “Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.”; Column 7: 51-54, “The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server.”*).

However, Hotti does not disclose:

- perform creates, alters, and drops of DDL objects associated with version  $i$  of the relational database;
- determining a set  $C$  of DML scripts that, when executed, modify DML objects that are associated with both version  $i$  and version  $j$  of the relational database, but that differ between version  $i$  and version  $j$  of the relational database;
- determining a set  $D$  of DML drop scripts that, when executed, drop DML objects that are associated with version  $i$  of the relational database, but that are not associated with version  $j$  of the relational database; and

- generating an upgrade file comprising a union of sets  $A$ ,  $B$ ,  $C$ , and  $D$   
 $(A \cup B \cup C \cup D)$ .

Official Notice is taken that it is old and well-known within the computing art to include create, alter, and drop commands in a DDL/DML script. A query language often provides these basic commands to manipulate a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include create, alter, and drop commands in a DDL/DML script. The modification would be obvious because one of ordinary skill in the art would be motivated to manipulate a database object.

Baisley discloses:

- determining a set  $C$  of DML scripts that, when executed, modify DML objects that are associated with both version  $i$  and version  $j$  of the relational database, but that differ between version  $i$  and version  $j$  of the relational database (*see Column 5: 38-45, "It is possible that a conflict may result if the same object or association is modified in both lines of development. Note that there is a conflict between model versions 33 and 34 where the attribute A.X=2 in model version 33 and A.X=1 in model version 34. The merge method of the present invention allows the user to specify if they want to keep the target value (version 34, A.X=1) or the source value (version 33, A.X=2). "*); and

- generating an upgrade file comprising a union of sets  $A$ ,  $B$ ,  $C$ , and  $D$   
 $(A \cup B \cup C \cup D)$  (*see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- determining a set  $C$  of DML scripts that, when executed, modify DML objects that are associated with both version  $i$  and version  $j$  of the relational database, but that differ between version  $i$  and version  $j$  of the relational database; and
- generating an upgrade file comprising a union of sets  $A$ ,  $B$ ,  $C$ , and  $D$   
( $A \bigcup B \bigcup C \bigcup D$ ).

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – *Column 1: 21-24*).

As per **Claim 14**, the rejection of **Claim 13** is incorporated; however, Hotti does not disclose:

- iteratively extracting sets  $M_k$ , each comprising zero or more filenames from metadata associated with a  $k^{th}$  version of the relational database, where  $i < k \leq j$ , the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $k-1$  of the relational database to version  $k$  of the relational database; and
- determining the set  $A$  as the union of sets  $M_{i+1}, M_{i+2}, \dots, M_j$   
( $A = M_{i+1} \bigcup M_{i+2} \bigcup \dots \bigcup M_j$ ).

Baisley discloses:

- iteratively extracting sets  $M_k$ , each comprising zero or more filenames from metadata associated with a  $k^{\text{th}}$  version of the relational database, where  $i < k \leq j$ , the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $k-1$  of the relational database to version  $k$  of the relational database (*see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. "*); and

- determining the set  $A$  as the union of sets  $M_{i+1}, M_{i+2}, \dots, M_j$  ( $A = M_{i+1} \bigcup M_{i+2} \bigcup \dots \bigcup M_j$ ) (*see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- iteratively extracting sets  $M_k$ , each comprising zero or more filenames from metadata associated with a  $k^{\text{th}}$  version of the relational database, where  $i < k \leq j$ , the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version  $k-1$  of the relational database to version  $k$  of the relational database; and

- determining the set  $A$  as the union of sets  $M_{i+1}, M_{i+2}, \dots, M_j$

$$(A = M_{i+1} \bigcup M_{i+2} \bigcup \dots \bigcup M_j).$$

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – *Column 1: 21-24*).

As per **Claim 15**, the rejection of **Claim 13** is incorporated; however, Hotti does not disclose:

- determining a set  $E$  of DML scripts that when executed:
  - perform alters of DML objects associated with version  $i$  and version  $j$  of the relational database, but that differ between version  $i$  and version  $j$  of the relational database; and

- perform creates of DML objects that are associated with version  $j$  of the relational database but that are not associated with version  $i$  of the relational database; and

- determining set  $B$  as the difference between sets  $E$  and  $C$  ( $B = E - C$ ).

Official Notice is taken that it is old and well-known within the computing art to perform alters and creates of DML objects. A query language often provides “create” and “alter” commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley discloses:

- determining set  $B$  as the difference between sets  $E$  and  $C$  ( $B = E - C$ ) (see *Column 5: 47-67*, “*TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34.*”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- determining set  $B$  as the difference between sets  $E$  and  $C$  ( $B = E - C$ ).

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – *Column 1: 21-24*).

As per **Claim 16**, the rejection of **Claim 15** is incorporated; however, Hotti does not disclose:

- iteratively determining a set  $P_x$  of DML scripts that when executed will upgrade DML objects from version  $x - 1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$ ;
- iteratively determining a set  $N_x$  of DML scripts that when executed will drop DML objects that are associated with version  $x - 1$  of the relational database but that are not associated with version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 2$  to  $j$ ;
- iteratively determining a set  $M_x$  of DML scripts that when executed will upgrade DML objects from version  $i$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$ , and where:

$$\begin{aligned}M_{i+1} &= P_{i+1} \\M_{i+2} &= [M_{i+1} \bigcup P_{i+2}] - N_{i+2} \\M_{i+3} &= [M_{i+2} \bigcup P_{i+3}] - N_{i+3} \\&\dots \\M_j &= [M_{j-1} \bigcup P_j] - N_j; \text{ and}\end{aligned}$$

- determining set  $E = M_j$ .

Baisley discloses:

- iteratively determining a set  $P_x$  of DML scripts that when executed will upgrade DML objects from version  $x - 1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$  (see Figure 3; Column 5: 17-36, “*In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.*”);

- iteratively determining a set  $N_x$  of DML scripts that are associated with version  $x - 1$  of the relational database but that are not associated with version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 2$  to  $j$  (see Figure 3; Column 5: 17-36, “*In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.*”);

- iteratively determining a set  $M_x$  of DML scripts that when executed will upgrade DML objects from version  $i$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$ , and where:

$$\begin{aligned}M_{i+1} &= P_{i+1} \\M_{i+2} &= [M_{i+1} \bigcup P_{i+2}] - N_{i+2} \\M_{i+3} &= [M_{i+2} \bigcup P_{i+3}] - N_{i+3} \\&\dots \\M_j &= [M_{j-1} \bigcup P_j] - N_j\end{aligned}$$

(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two."); and

- determining set  $E = M_j$  (see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.").

Official Notice is taken that it is old and well-known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- iteratively determining a set  $P_x$  of DML scripts that when executed will upgrade DML objects from version  $x - 1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$ ;

- iteratively determining a set  $N_x$  of DML scripts that when executed will drop DML objects that are associated with version  $x-1$  of the relational database but that are not associated with version  $x$  of the relational database, where  $x$  varies incrementally from  $i+2$  to  $j$ ;
- iteratively determining a set  $M_x$  of DML scripts that when executed will upgrade DML objects from version  $i$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from  $i+1$  to  $j$ , and where:

$$M_{i+1} = P_{i+1}$$

$$M_{i+2} = [M_{i+1} \bigcup P_{i+2}] - N_{i+2}$$

$$M_{i+3} = [M_{i+2} \bigcup P_{i+3}] - N_{i+3}$$

...

$$M_j = [M_{j-1} \bigcup P_j] - N_j; \text{ and}$$

- determining set  $E = M_j$ .

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – Column 1: 21-24).

As per **Claim 17**, the rejection of **Claim 13** is incorporated; and Hotti further discloses:

- determining a set  $F_j$  of DML scripts that when executed, create DML objects associated with version  $j$  of the relational database (see *Column 2: 9-10*, “*“Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.*”; *Column 4: 49-58*, “*Schema scripts can also include DML (Data Manipulation*

*Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ".*

However, Hotti does not disclose:

- determining a set  $E$  of DML scripts that when executed:
  - perform alters of DML objects associated with version  $i$  and version  $j$  of the

relational database, but that differ between version  $i$  and version  $j$  of the relational database;

and

- perform creates of DML objects that are associated with version  $j$  of the relational database but that are not associated with version  $i$  of the relational database; and
- determining set  $C$  as the intersection of set  $E$  and set  $F_j$  ( $C = E \cap F_j$ ).

Official Notice is taken that it is old and well-known within the computing art to perform alters and creates of DML objects. A query language often provides “create” and “alter” commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley discloses:

- determining set  $C$  as the intersection of set  $E$  and set  $F_j$  ( $C = E \cap F_j$ ) (see Column 5: 47-67, “*TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34.*”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- determining set  $C$  as the intersection of set  $E$  and set  $F_j$  ( $C = E \cap F_j$ ).

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – *Column 1: 21-24*).

As per **Claim 18**, the rejection of **Claim 17** is incorporated; however, Hotti does not disclose:

- extracting a set  $M_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;
- iteratively extracting a set  $M_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$ ;
- iteratively extracting a set  $B_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$ ; and
- determining the set  $F_j$  by determining:

$$\begin{aligned}F_2 &= [M_1 \bigcup M_2] - B_2, \\F_3 &= [F_2 \bigcup M_3] - B_3, \\F_4 &= [F_3 \bigcup M_4] - B_4, \\&\dots \\F_j &= [F_{j-1} \bigcup M_j] - B_j.\end{aligned}$$

Baisley discloses:

- extracting a set  $M_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version (see Figure 3; Column 5: 17-36, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model.");

- iteratively extracting a set  $M_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$  (see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. ");

- iteratively extracting a set  $B_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$  (see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new

*branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. "); and*

- determining the set  $F_j$  by determining:

$$F_2 = [M_1 \bigcup M_2] - B_2,$$

$$F_3 = [F_2 \bigcup M_3] - B_3,$$

$$F_4 = [F_3 \bigcup M_4] - B_4,$$

...

$$F_j = [F_{j-1} \bigcup M_j] - B_j$$

*(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ").*

Official Notice is taken that it is old and well-known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- extracting a set  $M_1$  comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

- iteratively extracting a set  $M_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$ ;
- iteratively extracting a set  $B_x$  comprising zero or more filenames from metadata associated with version  $x$  of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version  $x-1$  of the relational database to version  $x$  of the relational database, where  $x$  varies incrementally from 2 to  $j$ ; and
- determining the set  $F_j$  by determining:

$$F_2 = [M_1 \bigcup M_2] - B_2,$$

$$F_3 = [F_2 \bigcup M_3] - B_3,$$

$$F_4 = [F_3 \bigcup M_4] - B_4,$$

...

$$F_j = [F_{j-1} \bigcup M_j] - B_j.$$

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 19**, the rejection of **Claim 13** is incorporated; however, Hotti does not disclose:

- determining a set  $E$  of DML scripts that when executed:  
- perform alters of DML objects associated with version  $i$  and version  $j$  of the relational database, but that differ between version  $i$  and version  $j$  of the relational database; and

- perform creates of DML objects that are associated with version  $j$  of the relational database but that are not associated with version  $i$  of the relational database;  
- iteratively determining a set  $F_x$  of DML scripts that when executed, drop DML objects associated with version  $x - 1$  of the relational database that are not associated with version  $x$  of the relational database, where  $x$  varies incrementally from  $i + 1$  to  $j$ ;  
- determining a set  $G$  as the union of sets  $F_i, F_{i+1}, F_{i+2}, \dots, F_j$   
( $G = F_i \bigcup F_{i+1} \bigcup \dots \bigcup F_{i+2}$ ); and  
- determining set  $D$  as the difference between set  $G$  and set  $E$  ( $D = G - E$ ).

Official Notice is taken that it is old and well-known within the computing art to perform alters and creates of DML objects. A query language often provides "create" and "alter" commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley discloses:

- iteratively determining a set  $F_x$  of DML scripts that when executed, drop DML objects associated with version  $x - 1$  of the relational database that are not associated with

version  $x$  of the relational database, where  $x$  varies incrementally from  $i+1$  to  $j$  (see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two. ");

- determining a set  $G$  as the union of sets  $F_i, F_{i+1}, F_{i+2}, \dots, F_j$

( $G = F_i \bigcup F_{i+1} \bigcup \dots \bigcup F_{i+2}$ ) (see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. "); and

- determining set  $D$  as the difference between set  $G$  and set  $E$  ( $D = G - E$ ) (see Column 5: 47-67, "TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34. ").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include:

- iteratively determining a set  $F_x$  of DML scripts that when executed, drop DML objects associated with version  $x-1$  of the relational database that are not associated with version  $x$  of the relational database, where  $x$  varies incrementally from  $i+1$  to  $j$ ;

- determining a set  $G$  as the union of sets  $F_i, F_{i+1}, F_{i+2}, \dots, F_j$

( $G = F_i \bigcup F_{i+1} \bigcup \dots \bigcup F_{i+2}$ ); and

- determining set  $D$  as the difference between set  $G$  and set  $E$  ( $D = G - E$ ).

The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – Column 1: 21-24).

As per **Claim 20**, Hotti discloses:

- a processor (see *Column 3: 66 and 67*, “*A database system may include server computers, smart terminals, other terminals and network nodes.*”);
- a memory (see *Column 3: 66 and 67*, “*A database system may include server computers, smart terminals, other terminals and network nodes.*”);
- one or more data definition language (DDL) scripts, each associated with one or more versions of a relational database (see *Column 2: 9-10*, “*“Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.*”); *Column 4: 49-58*, “*Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.*”);
- one or more data manipulation language (DML) scripts, each associated with one or more versions of the relational database (see *Column 2: 9-10*, “*“Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.*”); *Column 4: 49-58*, “*Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.*”);
- a database schema version management structure definition (see *Column 1: 39-43*, “*“Database Schema” is the structure of a database system, described in a formal language supported by the database management system (DBMS). In a relational database, the schema*

*defines the tables, the fields in each table, and the relationships between fields and tables. ";*

*Column 2: 7-8, "Schema revision" is a snapshot version of a schema that is identifiable by logical name or version number. " and 14-15, "Schema script publication" is a system publication that contains the schema scripts of the database hierarchy. "); and*

- schema data associated with multiple versions of the relational database, the schema data organized according to the database schema version management structure definition (see

*Column 2: 14 and 15, "Schema script publication" is a system publication that contains the schema scripts of the database hierarchy. "; Column 6: 18-21, "The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system. ").*

However, Hotti does not disclose:

- an installation file generator stored in the memory and executed on the processor to apply laws of set theory to the schema data to generate a file comprising the one or more DDL scripts associated with a particular one of the multiple versions of the relational database, and the one or more DML scripts associated with the particular one of the multiple versions of the relational database.

Baisley discloses:

- an installation file generator stored in the memory and executed on the processor to apply laws of set theory to the schema data to generate a file comprising the one or more DDL scripts associated with a particular one of the multiple versions of the relational database, and the one or more DML scripts associated with the particular one of the multiple versions of the relational database (see Column 2: 14-19, "The method comprises the steps of building a first list

*as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include an installation file generator stored in the memory and executed on the processor to apply laws of set theory to the schema data to generate a file comprising the one or more DDL scripts associated with a particular one of the multiple versions of the relational database, and the one or more DML scripts associated with the particular one of the multiple versions of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – *Column 1: 21-24*).

As per **Claim 23**, Hotti discloses:

- a memory (see *Column 3: 66 and 67*, “*A database system may include server computers, smart terminals, other terminals and network nodes. ”*);
- a processor (see *Column 3: 66 and 67*, “*A database system may include server computers, smart terminals, other terminals and network nodes. ”*); and
- a database schema version management system stored in the memory, and executed on the processor to:
  - manage schema data associated with multiple versions of a relational database (see *Figure 2a: 234; Column 2: 7-8*, “*“Schema revision” is a snapshot version of a schema that*

*is identifiable by logical name or version number. "; Column 6: 17-20, "The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system. ").*

However, Hotti does not disclose:

- generate an installation file associated with any one of the multiple versions of the relational database.

Baisley discloses:

- generating an installation file comprising a union of the first set and the second set

*(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include generate an installation file associated with any one of the multiple versions of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database *(see Baisley – Column 1: 21-24).*

As per **Claim 24**, the rejection of **Claim 23** is incorporated; and Hotti further discloses:

- wherein the schema data identifies a script associated with a data definition language object of the relational database *(see Column 2: 9-10, ""Schema script"" is a script that creates a*

*schema or creates a new revision of an existing schema of a database node. "; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ").*

As per **Claim 25**, the rejection of **Claim 23** is incorporated; and Hotti further discloses:

- wherein the schema data identifies a script associated with a data manipulation language object of the relational database (*see Column 2: 9-10, "'Schema script'" is a script that creates a schema or creates a new revision of an existing schema of a database node. "; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ").*

As per **Claim 26**, the rejection of **Claim 23** is incorporated; however, Hotti does not disclose:

- wherein the database schema version management system is further configured to generate an installation file associated with an initial version of the relational database.

Baisley discloses:

- wherein the database schema version management system is further configured to generate an installation file associated with an initial version of the relational database (*see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists. ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include wherein the database schema version management system is further configured to generate an installation file associated with an initial version of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 27**, the rejection of **Claim 23** is incorporated; however, Hotti does not disclose:

- wherein the database schema version management system is further configured to generate an installation file associated with a non-initial version of the relational database.

Baisley discloses:

- wherein the database schema version management system is further configured to generate an installation file associated with a non-initial version of the relational database (*see Column 2: 14-19, “The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include wherein the database schema version management system is further configured to generate an installation file associated with a non-initial version of the relational database. The modification

would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 28**, the rejection of **Claim 23** is incorporated; however, Hotti does not disclose:

- wherein the database schema version management system is further configured to generate an upgrade file for upgrading one version of the relational database to another version of the relational database.

Baisley discloses:

- wherein the database schema version management system is further configured to generate an upgrade file for upgrading one version of the relational database to another version of the relational database (*see Column 2: 14-19, “The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include wherein the database schema version management system is further configured to generate an upgrade file for upgrading one version of the relational database to another version of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (*see Baisley – Column 1: 21-24*).

As per **Claim 29**, Hotti discloses:

- maintain schema data that identifies scripts associated with database objects of multiple sequential versions of a relational database (see Column 2: 9-10, *""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."* and 14-15, *""Schema script publication" is a system publication that contains the schema scripts of the database hierarchy."*; Column 4: 49-58, *"Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts."*).

However, Hotti does not disclose:

- generate an installation file associated with an initial version of the relational database by applying laws of set theory to the schema data to identify scripts associated with the database objects of the initial version of the relational database.

Baisley discloses:

- generate an installation file associated with an initial version of the relational database by applying laws of set theory to the schema data to identify scripts associated with the database objects of the initial version of the relational database (see Column 2: 14-19, *"The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley into the teaching of Hotti to include generate an installation file associated with an initial version of the relational database by applying laws of set theory to the schema data to identify scripts associated with the database objects of the initial version of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to define, track, and maintain objects, models, and versions in a database (see Baisley – Column 1: 21-24).

As per **Claim 30**, the rejection of **Claim 29** is incorporated; and Hotti further discloses:

- generate an installation file associated with a non-initial version of the relational database by applying laws of set theory to the schema data to identify:
  - scripts associated with data definition language (DDL) objects that are associated with the non-initial version of the relational database (see Column 2: 9-10, *“‘Schema script’ is a script that creates a schema or creates a new revision of an existing schema of a database node.”*; Column 4: 49-58, *“Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.”*); and
    - scripts associated with data manipulation language (DML) objects that are associated with the non-initial version of the relational database (see Column 2: 9-10, *“‘Schema script’ is a script that creates a schema or creates a new revision of an existing schema of a database node.”*; Column 4: 49-58, *“Schema scripts can also include DML (Data Manipulation*

*Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ").*

As per **Claim 31**, the rejection of **Claim 29** is incorporated; and Hotti further discloses:

- generate an upgrade file associated with an upgrade from a first, but not necessarily initial, version of the relational database to a second, later, but not necessarily immediately sequential, version of the relational database by applying laws of set theory to the schema data to identify:
  - data definition language (DDL) scripts associated with DDL objects of the relational database that have been created or modified between the first and second versions of the relational database (*see Column 2: 9-10, "Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node. "; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ";*
  - data manipulation language (DML) scripts associated with DML objects of the relational database that have been created between the first and second versions of the relational database (*see Column 2: 9-10, "Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node. "; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts. ";* and
    - DML scripts associated with DML objects of the relational database that have been modified between the first and second versions of the relational database (*see Column 2: 9-*

10, *“Schema script” is a script that creates a schema or creates a new revision of an existing schema of a database node.”; Column 4: 49-58, “Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.”).*

However, Hotti does not disclose:

- drop scripts associated with database objects that have been dropped and not re-created between the first and second versions of the relational database.

Official Notice is taken that it is old and well-known within the computing art to include a drop script as part of the DML. A query language often provides a “drop” command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include drop scripts associated with database objects that have been dropped and not re-created between the first and second versions of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

12. **Claim 7** is rejected under 35 U.S.C. 103(a) as being unpatentable over Hotti in view of Baisley as applied to Claim 6 above, and further in view of **Oracle8i Distributed Database Systems Release 8.1.5, 1999** (hereinafter Oracle1999).

As per **Claim 7**, the rejection of **Claim 6** is incorporated; however, Hotti and Baisley do not disclose:

- wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file.

Oracle1999 discloses:

- wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file (see *Page 2-13*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Oracle1999 into the teaching of Hotti to include wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file. The modification would be obvious because one of ordinary skill in the art would be motivated to define a stored procedure.

13. **Claims 9, 21, and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over Hotti in view of Baisley as applied to Claims 8 and 20 above, and further in view of **US 6,947,945 (hereinafter Carey)**.

As per **Claim 9**, the rejection of **Claim 8** is incorporated; however, Hotti and Baisley do not disclose:

- wherein the metadata comprises an XML file.

Carey discloses:

- wherein the metadata comprises an XML file (see *Column 1: 30-32, "An alternative data format to the tables found in an RDBMS is XML, which is a tag language for describing documents."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey into the teaching of Hotti to include wherein the metadata comprises an XML file. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize a future standard for information exchange between peer data stores, and between client visualization tools and data servers (*see Carey – Column 2: 15-17*).

As per **Claim 21**, the rejection of **Claim 20** is incorporated; however, Hotti and Baisley do not disclose:

- wherein the database schema version management structure definition comprises an XML schema definition.

Carey discloses:

- wherein the database schema version management structure definition comprises an XML schema definition (*see Column 1: 44-49, "XML schemas specify constraints on the structures and types of elements in an XML document." and "Other XML schema definitions are also being developed, such as XML Schema ... "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey into the teaching of Hotti to include wherein the database schema version management structure definition comprises an XML schema definition. The modification would be obvious because one of ordinary skill in the art would be motivated to specify constraints on the structures and types of elements in an XML document (*see Carey – Column 2: 44-49*).

As per **Claim 22**, the rejection of **Claim 21** is incorporated; however, Hotti and Baisley do not disclose:

- wherein the schema data is maintained in an XML file structured according to the XML schema definition.

Carey discloses:

- wherein the schema data is maintained in an XML file structured according to the XML schema definition (*see Column 1: 30-32, "An alternative data format to the tables found in an RDBMS is XML, which is a tag language for describing documents. "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey into the teaching of Hotti to include wherein the schema data is maintained in an XML file structured according to the XML schema definition. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize a future standard for information exchange between peer data stores, and between client visualization tools and data servers (*see Carey – Column 2: 15-17*).

#### ***Response to Arguments***

14. Applicant's arguments with respect to Claims 1, 13, 20, 23, and 29 have been considered, but are moot in view of the new ground(s) of rejection.

***In the remarks, Applicant argues that:***

a) While the cited portions of the reference may describe updating database schemas, the reference does not disclose automatically determining sets of scripts. Specifically, Applicant fails to see how the cited portions of Hotti disclose, "automatically determining a first set of data definition language (DDL) scripts associated with the particular version of the relational database;" and "automatically determining a second set of data manipulation language scripts associated with the particular version of the relational database," as recited in claim 1.

***Examiner's response:***

a) Examiner disagrees. Hotti clearly discloses:

- automatically determining a first set of data definition language (DDL) scripts associated with the particular version of the relational database (*see Column 2: 9-10, "Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."*; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts."); *Column 7: 51-54, "The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server."*); and
- automatically determining a second set of data manipulation language scripts associated with the particular version of the relational database (*see Column 2: 9-10, "Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."*; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation

*scripts. "; Column 7: 51-54, "The application master schema is updated by running the scripts of the new revision, 506. The scripts are found from the configuration management replica database of the server. ").*

***In the remarks, Applicant argues that:***

b) With reference to claims 3 and 4, the Examiner cites Hotti, column 6 lines 63-66, which states, "As part of the registrations, the identification data, e.g. schema name, of the new application database node is sent to the configuration management master database node." This cited portion of the reference indicates that a schema name may be sent to a master database node. However, there is nothing to indicate that the schema name is a filename, as recited in the claims. Furthermore, the cited portion of the reference does not disclose extracting anything (a filename in particular) from metadata.

***Examiner's response:***

b) Examiner disagrees. Hotti clearly discloses extracting a filename from metadata associated with the first version (see Column 6: 18-21, "The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system." and 53-55, "In step 302 the database schema is defined using configuration management application and stored to the schema management master database, step 303." and 63-67 to Column 7: 1, "As part of the registration, the identification data, e.g. schema name, of the new application database node is sent to the

*configuration management master database node. The newly created configuration management replica is then synchronized with its master database in step 308. ")).*

Note that the schemas are managed by the configuration management application (metadata). The process of synchronization between the configuration management replica database and the configuration management master database requires the identification of a schema (schema name). A schema is a file and thus, a schema name is the filename of the schema.

***In the remarks, Applicant argues that:***

c) It is important to note, however, that in column 2, lines 10-14, Baisley states, "These and other objects and advantages, which will become apparent as the invention is described in detail below, are provided by a computer-implemented method wherein a source version is to be merged into a target version of a model in an object oriented repository." As is well known to those skilled in the art, an object oriented repository differs from a relational database. Claim 13 is clearly directed to a method related to a relational database, not an object oriented database. Accordingly, Applicant respectfully submits that the combination of Hotti and Baisley is improper.

***Examiner's response:***

c) In response to Applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on

combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hotti in view of Baisley. Hotti is directed to a method related to a relational database. Baisley is directed to a method related to an object oriented database. The combination of Hotti and Baisley is proper because both Hotti and Baisley are analogous prior art. Hotti is in the same field as the Applicant's endeavor—namely, managing database schemas. Baisley is reasonably pertinent to the particular problem with which the Applicant was concerned—namely, generating a master file as a union of other files. Baisley is relied upon for its specific teaching of generating a master file as a union of other files. The combined teaching of Hotti and Baisley supports the conclusion that the claimed invention is directed to obvious subject matter.

***In the remarks, Applicant argues that:***

d) Furthermore, while the cited portion of Baisley may mention a union of two lists, the cited portion of Baisley does not disclose, "generating an upgrade file," as recited in claim 13. Rather, the cited portion of Baisley describes merging a source version of a model into a target version of a model in an object oriented repository. Applicant fails to see how Baisley's description of merging two versions of a model in an object oriented repository discloses, generating an upgrade file to upgrade version i of a relational database to version j of the relational database, where j > i, as recited in claim 13..

***Examiner's response:***

d) In response to Applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Note that a DDL or DML script is an upgrade file in itself. The claimed upgrade file is merely a consolidation of the various DDL and DML scripts that are required to upgrade a database to a newer version. Hotti discloses executing schema scripts (DDL and DML scripts) to upgrade the database schema version. However, Hotti does not disclose that the schema scripts are consolidated into one script file. Baisley discloses generating a history file as a union of the list of changes that occur in the source version and the list of changes that occur in the target version. In light of the teachings of the references, one of ordinary skill in the art would modify Hotti to consolidate the schema scripts into one script file to define, track, and maintain objects, models, and versions in a database.

Note that Applicant did not traverse the Examiner's assertion of Official Notice with regard to Claims 11, 13, 15-19, and 31. Therefore, the "old and well-known within the computing art" statement is taken to be admitted prior art because Applicant has failed to traverse the Examiner's assertion of Official Notice (see MPEP § 2144.03).

### ***Conclusion***

15. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



WEI ZHEN  
SUPERVISORY PATENT EXAMINER